

PLUSPAT INPUTS

11/30 PLUSPAT - (C) QUESTEL-ORBIT  
PN - JP9179754 A 19970711 [JP09179754]  
TI - (A) TASK MONITORING DEVICE AND ITS METHOD  
PA - (A) MITSUBISHI ELECTRIC CORP  
IN - (A) BABA YOSHIYUKI  
AP - JP33350895 19951221 [1995JP-0333508]  
PR - JP33350895 19951221 [1995JP-0333508]  
IC - (A) G06F-009/46 G06F-011/30 G06F-011/30 G06F-011/32 G06F-011/34  
STG - (A) Doc. Laid open to publ. Inspec.  
AB - PROBLEM TO BE SOLVED: To allow a task itself to recognize its stop time, to visually grasp the stop state of the task and to automatically cope with the execution delay of the task.  
- SOLUTION: An operating system(OS) part 2 for attaining a multi-task execution format is provided with a task stop allowable time setting means 2a, a stop time accumulating means 2b, a stop state provision means 2c, and a stop time excess reporting means 2d, and the OS part is also provided with a means for managing the stop time of a user definition task to be executed by a computer, and when the stop time exceeds allowable time, informing the task of the event. A system managing monitor 3 to be driven as a task on the computer is provided with a task stop record collecting means 3a, a stop state displaying means 2b, a bottleneck detecting means 3c, a fault informing means 3d, a task state transition displaying means 3e, and a task inputting means 3f. The monitor 3 enters the task stop state recorded by the OS part 2 by the use of an application interface and manages the performance and operation of a computer system.  
- COPYRIGHT: (C)1997, JPO

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-179754

(43) 公開日 平成9年(1997)7月11日

(51) Int.Cl. <sup>6</sup>	識別記号	序内整理番号	F I	技術表示箇所
G 06 F 11/30	3 0 5	7313-5B	G 06 F 11/30	3 0 5 A
		7313-5B		K
9/46	3 4 0		9/46	3 4 0 B
11/32		7313-5B	11/32	B
11/34		7313-5B	11/34	A

(21) 出願番号 特願平7-333508

(22)出願日 平成7年(1995)12月21日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 馬場 健之

東京都千代田区丸の内二丁目2番3号 三

## 菱電機株式会社内

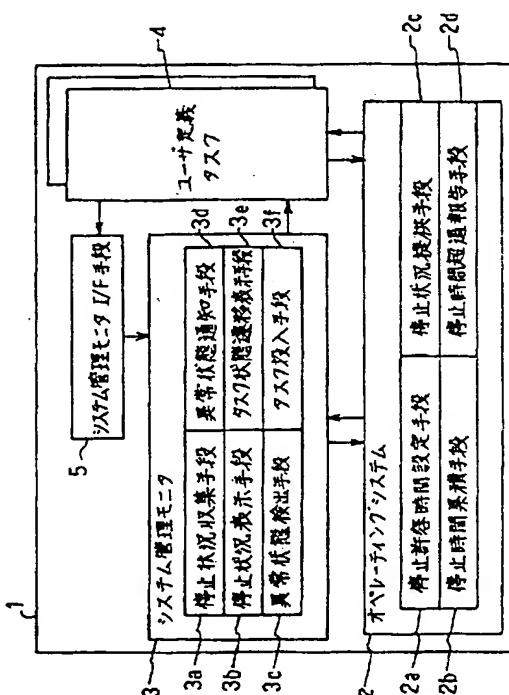
(74) 代理人 弁理士 宮田 金雄 (外3名)

(54) 【発明の名称】 タスク監視装置及びタスク監視方法

(57)【要約】

【課題】 タスクの停止時間をタスク自身が認識すること、タスクの停止状況を視覚的に把握すること、及びタスクの実行遅れに対して自動的に対処すること。

【解決手段】 マルチタスク実行形式を実現するOS部に、タスクの停止許容時間の設定、停止時間の累積、停止状況の提供、停止時間超過の報告の手段を設けると共に、OSに、計算機上で実行するユーザ定義タスクの停止時間を管理し、停止時間が許容時間を超過した場合に、タスクに事象通知する手段を設ける。また、計算機上で1タスクとして動作するシステム管理モニタに、タスクの停止記録の収集、停止状況の表示、ボトルネック検出、障害の通知、タスクの状態遷移の表示、タスク投入手段を設ける。システム管理モニタは、OSが記録したタスクの停止状況をアプリケーションインターフェースを用いて取り込み、計算機システムの性能管理及び運転管理を行う。



## 【特許請求の範囲】

【請求項1】 複数のタスクが並列に実行される計算機であって、タスクの実行を制御するオペレーティングシステムに、下記の要素を設けたことを特徴とするタスク監視装置

(a) タスク毎にそのタスクの停止時間の許容値を設定する停止許容時間設定手段、(b) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、(c) 前記停止時間累積手段で記録されたタスクの停止時間とそのタスクに対応する前記許容値とを比較する比較手段、

(d) 前記比較手段でタスクの停止時間が前記許容値を越えると判断したときには、前記タスクに停止時間が許容値を越えたことを報告する停止時間超過報告手段。

【請求項2】 複数のタスクが並列に実行される計算機であって、下記の要素を有する、タスクの実行を制御するオペレーティングシステムと、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、

下記の要素を有する、所定のインターバルで動作するシステム管理モニタと(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) 前記停止状況収集手段の収集したタスクの停止状況を基にタスクの負荷状況を度数分布で表示する停止状況表示手段、

を備えたことを特徴とするタスク監視装置。

【請求項3】 前記停止状況表示手段は、所定のインターバルにおけるタスクの負荷状況を表示することを特徴とする請求項2に記載のタスク監視装置。

【請求項4】 前記停止状況表示手段は、累積したタスクの負荷状況を表示することを特徴とする請求項2に記載のタスク監視装置。

【請求項5】 前記停止状況提供手段は、停止状況として停止状態中の情報も含めて提供することを特徴とする請求項2乃至請求項4のいずれかに記載のタスク監視装置。

【請求項6】 複数のタスクが並列に実行される計算機であって、下記の要素を有する、タスクの実行を制御するオペレーティングシステムと、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、(b) 前記停止時間累積手段で記録されたタスクの停止状況を

提供する停止状況提供手段、

下記の要素を有する、所定のインターバルで動作するシステム管理モニタと(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) タスク毎にそのタスク異常状態を定義した異常状態定義情報、(c) 前記停止状況と前記異常状態定義情報を基にタスクの異常状態を検出する異常状態検出手段と、(d) 異常状態検出手段で異常と判断したときには、前記タスクに対して異常であることを通知する異常通知手段、

を備えたことを特徴とするタスク監視装置。

【請求項7】 複数のタスクが並列に実行される計算機であって、下記の要素を有する、タスクの実行を制御するオペレーティングシステムと、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、

下記の要素を有する、所定のインターバルで動作するシステム管理モニタと(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) タスク毎にそのタスク異常状態を定義した異常状態定義情報、(c) 前記停止状況と前記異常状態定義情報を基にタスクの異常状態を検出する異常状態検出手段と、

(d) 異常状態検出手段で異常と判断したときには、この異常状態を処理するタスクを起動するタスク投入手段、

を備えたことを特徴とするタスク監視装置。

【請求項8】 前記異常状態定義情報をユーザタスクから更新するシステム管理モニタインタフェース手段を設けたことを特徴とする請求項6または請求項7に記載のタスク監視装置。

【請求項9】 前記システム管理モニタに前記タスクの停止状況を基にタスクの遷移状態を表示するタスク状態遷移表示手段を設けたことを特徴とする請求項6乃至請求項8のいずれかに記載のタスク監視装置。

【請求項10】 複数のタスクが並列に実行される計算機であって、下記の要素を有する、タスクの実行を制御するオペレーティングシステムと、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、

ユーザタスクが設定するタスクの投入情報を保持するシステム管理モニタインタフェース手段と、

下記の要素を有する、所定のインターバルで動作するシ

システム管理モニタと (a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) 前記停止状況収集手段の収集したタスクの停止状況を基にタスクの投入の可否を判断する投入可否判断手段と、(c) 前記タスク投入可否判断手段で投入可と判断したときに前記システム管理モニタインタフェース手段から得たタスク投入情報を基にタスクを投入するタスク投入手段、

を備えたことを特徴とするタスク監視装置。

【請求項11】 下記の工程を有するタスク監視方法  
(a) タスク毎にそのタスクの停止時間の許容値を設定する工程、(b) 前記タスクが停止したとき、その停止時刻、停止原因を記録する工程、(c) 前記タスクが停止状態から回復したとき、停止時間及び停止原因と停止時間の累計値を記録する工程、(d) 前記停止時間と前記許容値とを比較する比較工程、(e) 前記比較工程で停止時間が許容値を越えると判断したときには、前記タスクに対して許容時間を越える停止が発生したことを報告する工程。

【請求項12】 下記の工程を有するタスク監視方法  
(a) タスクが停止したとき、その停止時刻、停止原因を記録する工程、(b) 前記タスクが停止状態から回復したとき、停止時間及び停止原因と停止時間の累計値等のタスクの停止状況を記録する工程、(c) 前記停止状況を基にタスクの負荷状況を度数分布で表示する工程。

【請求項13】 下記の工程を有するタスク監視方法  
(a) タスク毎にそのタスクの異常状態を定義する工程、(b) 前記タスクが停止したとき、その停止時刻、停止原因を記録する工程、(c) 前記タスクが停止状態から回復したとき、停止時間及び停止原因と停止時間の累計値等のタスクの停止状況を記録する工程、(d) 前記停止情報を基に定義された前記異常状態の発生を検出する異常状態検出工程、(e) 前記異常状態検出工程で異常を検出するとこの異常状態を処理する工程。

【請求項14】 前記異常状態を処理する工程は、異常状態に該当するタスクに報告する工程であることを特徴とする請求項14に記載のタスク監視方法。

【請求項15】 前記異常状態を処理する工程は、この異常状態を処理するタスクを起動する工程であることを特徴とする請求項14に記載のタスク監視方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、マルチタスク実行方式の計算機において、タスクの停止時間の管理を行い、タスクに処理の時間的遅れを認識させる事を可能にすると共に、計算機システムの負荷状況の監視を可能にするものである。更に、計算機システムの性能管理及び運転管理を行うことを可能にする技術に関するものである。

【0002】

【従来の技術】従来のマルチタスク実行方式の計算機では、計算機の性能を評価するための各種評価ツールが作成され、提供されている。例えば、UNIXのpsコマンドでは、オペレーティングシステム(以降、OSと称す)の管理下で実行されるプログラムの個別環境であるプロセスという実行単位毎に、OSによる管理情報を表示する。同じくUNIXでは、計算機でのシステム資源の使用状況を把握するための評価ツールとして、vmsstatコマンドが提供されている。vmsstatコマンドでは、システム資源の過剰利用によるレスポンス低下を検出するためのデータを表示する。また、マルチタスク実行環境においては、トランザクション処理において、処理のレスポンスを保証するためにトランザクション処理モニタを用いている。

【0003】psコマンドでは、プロセス(プログラム)が停止している原因を表すイベント情報を、例えば図13に示すコマンド出力でのWCHAN欄に、ソフトウェアのアドレス情報で表示する。ここで、WCHAN欄にアドレス表示がないプロセスは、実行可能状態にあるもの、あるいは実際はOSの一部であるプログラムである。psコマンド出力のWCHAN欄に表示するアドレス情報等の表示データは、OSがプロセス毎に割り当てるテーブル内に管理している。psコマンドは、テーブル内のデータを得るためのアプリケーション・インターフェース(以下、API)を用い、これによって得たデータを表示する。しかしながら、この情報をタスク自身で判断することはできなかった。

【0004】また、vmsstatコマンドでは、図14の様な出力が得られる。出力において、procsの項目ではCPU待ちの状態、memoryの項目ではメモリの使用状況、pageの項目ではページング処理の状況、faultの項目では割り込みの発生状況、cpuの項目ではCPUの利用状況を表している。これらの情報は、OSによってOS内部のデータとして管理されている。vmsstatコマンドにおいても、これらOS内部のデータを得るためにAPIを実行し、これによって得たデータを表示する。

【0005】また、オンライン・トランザクション処理システムでは、トランザクション処理モニタを使って計算機システムに投入するトランザクションを整理することで、システムのスループットを向上させる手段が採られている。このトランザクション処理モニタでは、実行リクエストをキューに貯めた後に、データベース・アプリケーションに渡すキューイング機能、優先度に従って順番に処理するスケジューリング機能を備えている。この機能により、トランザクション処理モニタを用いた計算機システムでは、処理のレスポンス時間を保証している。

【0006】しかしながら、以上のような情報は、全てOS自身でしか判断できず、タスク自身がこの情報を使

用することができなかつた。また、この情報を取り込むインターフェースを設けたとしても、その情報を判断することはできなかつた。

【0007】

【発明が解決しようとする課題】計算機におけるタスクの実行では、外乱によってタスクの実行遅れが生じ得る。ここでいう外乱とは、タスクの実行を継続できなくなるイベント全般を指している。このイベントには例えば、各種 I/O の終了待ち、メモリや CPU 等の計算機資源待ち、マルチタスク実行下での他タスクとの通信、他計算機上のタスクとの通信、タスク自身が設定した停止等がある。このため、停止時間の管理を行っていない計算機上のタスクは、外乱による実行遅れを知ることができない。

【0008】また、従来のシステム評価ツールでは、OS がタスクの停止時間を管理していないため、外乱が引き起こすレスポンス低下やシステムハング等の原因を即座に把握することが困難であった。例えば、I/O においてリトライが頻発してシステムのレスポンス低下が起きた時には、1 回の I/O に費やした時間によって異常を検知するのではなく、I/O のリトライ回数もしくは I/O サービスを提供するソフトウェアのログによって異常を検知する必要があった。

【0009】また、実行リクエストのキューを管理する従来のトランザクション処理モニタでは、1 個の実行リクエストを増やすことが、OS の処理でのオーバーヘッドによって、タスクにどれだけの遅れを生じるかを把握することが困難であった。

【0010】本発明は上記のような問題点を解消するためになされたもので、マルチタスク実行方式の計算機において、以下の機能を得ることを目的とする。

(1) マルチタスク実行を実現する OS において、タスクに実行遅れを通知するための処理をするようにして、タスク自身で実行遅れを認識することができるようにして、タスク自身での回復処理を自動化することを可能にする。

(2) システム管理モニタを 1 つのタスクとして実現して、一定時間毎にシステムで実行中のタスクが停止している原因毎の停止時間分布を得て、計算機システムのボトルネック解析を容易にする。

(3) また、システム管理モニタでは、任意タスクにおける停止原因毎の停止時間分布も得ることを可能にする。

(4) また、システム管理モニタでは、任意タスクにおける停止時間を監視し、停止時間が許容値を超えた時には、利用者が定義するアクションを自動的に実行する。

(5) また、システム管理モニタでは、タスクの停止原因と停止時間に注目した、タスクの状態遷移情報を提供する。

(6) また、システム管理モニタでは、実行中のタスク

の停止時間に基づいて負荷状況を判断して、負荷状況に応じて計算機システムに投入するタスクを制御する。

【0011】

【課題を解決するための手段】本発明に係わるタスク監視装置は、複数のタスクが並列に実行される計算機であって、タスクの実行を制御するオペレーティングシステムに、下記の要素を設けるようにしたものである。

(a) タスク毎にそのタスクの停止時間の許容値を設定する停止許容時間設定手段、(b) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、(c) 前記停止時間累積手段で記録されたタスクの停止時間とそのタスクに対応する前記許容値とを比較する比較手段、

(d) 前記比較手段でタスクの停止時間が前記許容値を越えると判断したときには、前記タスクに停止時間が許容値を越えたことを報告する停止時間超過報告手段。

【0012】また、複数のタスクが並列に実行される計算機であって、タスクの実行を制御するオペレーティングシステムに、下記の要素を設け、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、

(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、所定のインターバルで動作するシステム管理モニタに、下記の要素を設けるようにしたものである。

(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) 前記停止状況収集手段の収集したタスクの停止状況を基にタスクの負荷状況を度数分布で表示する停止状況表示手段。

【0013】また、前記停止状況表示手段は、所定のインターバルにおけるタスクの負荷状況を表示するようにしたものである。

【0014】また、前記停止状況表示手段は、累積したタスクの負荷状況を表示するようにしたものである。

【0015】また、前記停止状況提供手段は、停止状況として停止状態中の情報を含めて提供するようにしたものである。

【0016】また、複数のタスクが並列に実行される計算機であって、タスクの実行を制御するオペレーティングシステムに、下記の要素を設け、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、

(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、所定のインターバル

で動作するシステム管理モニタに、下記の要素を設けるようにしたものである。

(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) タスク毎にそのタスク異常状態を定義した異常状態定義情報、(c) 前記停止状況と前記異常状態定義情報を基にタスクの異常状態を検出する異常状態検出手段と、(d) 異常状態検出手段で異常と判断したときには、前記タスクに対して異常であることを通知する異常通知手段。

【0017】また、複数のタスクが並列に実行される計算機であって、タスクの実行を制御するオペレーティングシステムに、下記の要素を設け、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、

(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、所定のインターバルで動作するシステム管理モニタに、下記の要素を設けるようにしたものである。

(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) タスク毎にそのタスク異常状態を定義した異常状態定義情報、(c) 前記停止状況と前記異常状態定義情報を基にタスクの異常状態を検出する異常状態検出手段と、(d) 異常状態検出手段で異常と判断したときには、この異常状態を処理するタスクを起動するタスク投入手段。

【0018】また、前記異常状態定義情報をユーザタスクから更新するシステム管理モニタインターフェース手段を設けるようにしたものである。

【0019】また、前記システム管理モニタに前記タスクの停止状況を基にタスクの遷移状態を表示するタスク状態遷移表示手段を設けるようにしたものである。

【0020】また、複数のタスクが並列に実行される計算機であって、タスクの実行を制御するオペレーティングシステムに、下記の要素を設け、(a) タスクの動作状態が遷移する度に、その遷移状態に応じて、タスク毎にそのタスクの停止した原因、停止時刻、停止時間及び、タスクの停止原因別にその停止原因並びに停止時間の累積値等の停止状況を記録する停止時間累積手段、

(b) 前記停止時間累積手段で記録されたタスクの停止状況を提供する停止状況提供手段、ユーザタスクが設定するタスクの投入情報を保持するシステム管理モニタインターフェース手段を備え、所定のインターバルで動作するシステム管理モニタに、下記の要素を設けるようにしたものである。

(a) 前記停止状況提供手段の提供する停止状況を収集する停止状況収集手段、(b) 前記停止状況収集手段の収集したタスクの停止状況を基にタスクの投入の可否を判断する投入可否判断手段と、(c) 前記タスク投入可

否判断手段で投入可と判断したときに前記システム管理モニタインターフェース手段から得たタスク投入情報を基にタスクを投入するタスク投入手段。

【0021】また、本発明によるタスク監視方法は、下記の工程を有するものである。

(a) タスク毎にそのタスクの停止時間の許容値を設定する工程、(b) 前記タスクが停止したとき、その停止時刻、停止原因を記録する工程、(c) 前記タスクが停止状態から回復したとき、停止時間及び停止原因と停止時間の累積値を記録する工程、(d) 前記停止時間と前記許容値とを比較する比較工程、(e) 前記比較工程で停止時間が許容値を越えると判断したときには、前記タスクに対して許容時間を越える停止が発生したことを報告する工程。

【0022】また、下記の工程を有するものである。

(a) タスクが停止したとき、その停止時刻、停止原因を記録する工程、(b) 前記タスクが停止状態から回復したとき、停止時間及び停止原因と停止時間の累積値等のタスクの停止状況を記録する工程、(c) 前記停止状況を基にタスクの負荷状況を度数分布で表示する工程。

【0023】また、下記の工程を有するものである。

(a) タスク毎にそのタスクの異常状態を定義する工程、(b) 前記タスクが停止したとき、その停止時刻、停止原因を記録する工程、(c) 前記タスクが停止状態から回復したとき、停止時間及び停止原因と停止時間の累積値等のタスクの停止状況を記録する工程、(d) 前記停止情報を基に定義された前記異常状態の発生を検出する異常状態検出手段、(e) 前記異常状態検出手段で異常を検出するとこの異常状態を処理する工程。

【0024】また、異常状態を処理する工程を、この異常状態に該当するタスクに報告する工程としたものである。

【0025】また、異常状態を処理する工程を、この異常状態を処理するタスクを起動する工程としたものである。

【0026】

【発明の実施の形態】図1は、本発明によるタスク監視装置の一実施の形態の全体構成を示す図である。図において、1はタスク監視装置を構成するソフトウェアの全体構成を示し、2はオペレーティングシステム(以下、OSと称す)である。OS2はタスクの停止してもよい停止時間を設定する許容時間設定手段2a、停止したタスクの停止時間及び停止原因累積手段2b、タスク停止状況提供手段2c、及び停止時間超過の報告手段2dを有している。3は所定の周期で動作するシステム管理モニタで、OS2上で実行される1つのタスクとして実現されている。このシステム管理モニタ3はタスクの停止状況収集手段3a、停止状況表示手段3b、タスクの異常状態検出手段3c、異常状態通知手段3d、タスク状態遷移表示手段3e及びタスクを起動するタスク投入手

段3fを有している。4はOS2の管理下で実行されるユーザ定義のタスク（以下、タスクと称す）で、5はユーザ定義タスク4とシステム管理モニタ3とのインターフェースをとるシステム管理モニタインタフェース手段（以降、システム管理モニタI/F手段と称す）である。

【0027】実施の形態1. この実施の形態1では、OS2がタスク4の停止時間を管理して、停止時間超過時にタスク4に事象通知を行う例について説明する。タスク4においては、OS2に対してタスク固有の停止時間の許容値を停止許容時間設定手段2aにより設定することが可能になっており、タスク4の停止時間がこの許容値を超過した場合、OS2の有する停止時間超過報告手段2dによってタスク4が検知可能な手段で事象通知を受けることができる。タスクへの事象通知方法の例としては、例えばUNIX等のシグナル等の手段がある。

【0028】図2は、実施の形態1におけるタスクの停止時間の累積処理と停止時間の報告時期を説明する図で、この図を参照しながら説明する。図2に示すグラフ10は、1つのタスクの状態遷移を時系列で表している。縦軸ではタスクの状態、横軸では時間の流れを表している。11aの状態は、注目しているタスク4がCPU（以後プロセッサとも称す）によって実行されている状態である。11bの状態における時刻で12aの状態に遷移するのは、タスク4がプログラム内で資源を要求した等で、停止した状態を表している。11cの状態の時刻で13aの状態に遷移するのは、タスク4がCPUの利用を制限された場合を表している。12aから12bの期間は、タスク4が要求した資源を獲得するまでに、タスク4が停止した時間を表している。また、13aから13bの期間は、タスク4がCPUによる実行権を持つまで、タスク4が停止した時間を表している。11dからは、タスク4が再び実行されている状態を表している。

【0029】図3は、OS2がタスク4毎に管理しているタスク管理テーブル20とタスクの停止状況をトレースするトレースバッファ30を示している。タスク管理テーブル20は、タスク4毎にOS2が割り当てるタスク管理テーブルであり、ここにはOS2がタスク4に関する管理データを格納されている。即ち、タスク4が最後に停止した時刻21、最後に停止した原因22、許容停止時間23、総停止時間24、停止原因別の許容停止時間25、停止原因別の累積回数26、停止原因別の累積時間27等のタスクの停止状況が格納される。

【0030】また、OS2のデータ領域の一部に設けられたトレースバッファ30にも、タスクの停止状況をOS2が書き込んで行く。40はOS2がタスクの停止状況を記録したタスク管理テーブル20とトレースバッファ30からデータを取り出すAPIであり、タスクプログラム（タスク）において実行可能になっている。シス

テム管理モニタ3は、このAPI40を用いてOS2から種々のデータを受け取る。

【0031】OS2は、タスク4への事象通知のために、タスク毎に停止原因別の停止回数及び停止時間を累積して管理する。タスクの停止事象の管理は、OS2がタスク4の切替えを行う際のコンテキスト保存処理（図2における12a, 13aでの処理）及びコンテキスト回復処理（図2における12b, 13bでの処理）において、計算機起動時からの経過時間を記録しているタイマを用いて行う。また、OS2は同時に、その時点でのタイムスタンプ31、回復処理を施したタスク4の識別子32、停止時間33、停止原因34等のトレースデータを記憶装置上に確保したトレースバッファ30上に書き出す。トレースバッファ30への書き出しは、回帰的に行い、システム管理モニタ3が時系列的にデータを得ることを可能にさせる。もし、タスク4における停止時間があらかじめ設定された許容値を超過した場合、OS2は図2における11dのタイミングにおいてタスク4で検知可能な方法を用いて、即ち停止時間超過報告手段2dにより事象通知する。

【0032】以下に、この実施の形態1におけるタスク4の停止時間の管理の処理の詳細を図4に示した流れ図を用いて説明する。ステップS101において、タスク4が停止した原因が、事象待ちか否かを判断して、その結果により処理が異なる。なお、ここでの事象待ちとは、タスク4の実行を停止する要因となるもの、即ちCPU利用待ち以外のものである。停止した原因が事象待ちでない場合には、ステップS120の処理でコンテキストを保存して事象待ちに関係する処理を行わない。

【0033】ステップS102では、タスク切替えを前提として今まで実行していたタスク4の実行環境を保存する処理を行う。ステップS103では、OS2が管理するタスク管理テーブル20に、最後に停止した時刻21（タスクが事象待ちを開始した時間情報）と最後に停止した原因22を記録する処理を行う。また、トレースバッファ30には、タスク4が事象待ちを開始した時刻情報（タイムスタンプ）31、タスク識別子32、停止原因34等のタスクの停止状況を書き出す。ステップS104では、タスク4が事象待ちにより停止していることを表している。マルチタスク実行形式のOS2では、この後、停止事象が解決されたタスク4に対して処理を行う。

【0034】ステップS105において、事象待ちが解決したタスク4があるか否かを判断して、その結果により処理が異なる。なお、このステップS105には、事象待ちの終了割り込み処理でも入ってくる（図5のBからの流れ）。

【0035】ステップS106において、事象待ちが解決したタスク4を選択する処理を行う。次に、ステップS107において、停止原因が解決されて実行を再開す

る時のロギング処理を行う。ここでは、OS2が管理するタスク管理テーブル20の、総停止時間24と停止原因別の累積回数25、停止原因別の累積時間27を更新する。また、トレースバッファ30には、タスク4での事象待ちが解決した時刻情報（タイムスタンプ）31、タスク識別子32、停止時間33（タイムスタンプ31から最後に停止した停止時刻21を減算して求める）、停止原因34を書き出す。

【0036】次に、ステップS108において、タスク4がCPU待ちを開始した停止情報に関して、ロギング処理を行う。ここでの処理は、ステップS103における処理と同様であるが、ログ中の停止原因の情報が異なる。次に、ステップS109において、該当するタスク4をCPU待ちのためにキューイングする処理を行う。次に、ステップS110において、タスク4を停止させていた事象が解決したタスク4があるか否かを判断して、その結果により処理が異なる。もし、事象待ちが解決したタスク4がある場合には、ステップS105から処理を実行し、そうでない場合には、ステップS111に移行する。ステップS111では、CPU待ちをしているタスク4が存在するか否かを判断して、その結果により処理が異なる。CPU待ちしているタスク4が存在しない場合には、CPUはアイドル状態になる。このアイドル状態が解決するのは、各種の割り込みを受けた時で、その時はB（ステップS105：事象待ち解決の場合）またはC（ステップS111：一定時間後のスケジューリング割り込み）から処理を再開する。

【0037】CPU待ちをしているタスク4がある場合には、ステップS112に移行して、CPU待ちキューから実行タスク選択する処理を行う。次に、ステップS113において、停止原因が解決されて実行を再開するためのロギング処理を行う。ここでの処理は、ステップS107の処理と同様であるが、ログ中の停止原因の情報が異なる。次に、ステップS114において、CPUの実行権を得たタスク4に対する、タスクの実行環境を回復する処理を行う。その後、ステップS115において、タスク4の停止時間が許容時間を超過したかどうかの判断処理を行う。停止時間が許容値を超過していた場合には、OS2はステップS130の処理でタスク4に対して事象通知を行う。

【0038】以上のように、この実施の形態1においては、OS2におけるタスク4の停止時間管理において、タスク4の停止前後において、停止原因と停止時間に関する情報を記録している。更に、もし、タスク4の停止時間が許容値を超過していた場合、OS2はタスクに対して事象通知するようにしたので、タスク自身で停止時間が超過したことを認識できるようになる。

【0039】実施の形態2、この実施の形態2では、本発明によるタスク監視装置におけるシステム管理モニタ3での負荷状況表示について説明する。本発明において

は、計算機上で1つのタスクとして実現されて動作するシステム管理モニタ3が、計算機上で実行されている全てのタスク4の停止事象について、以下の情報を提供する。

(a) 特定のタスクにおける、停止原因別の停止時間の度数分布。

(b) 計算機単体の全タスクにおける、停止原因別の停止時間の度数分布。なお、タスクの停止原因はOSの実現方法に依存する。例えば、UNIXにおいては、以下のようないくつかの原因がある。

・各種入出力装置の終了待ち。

例) 磁気ディスク装置、端末装置、ネットワーク装置等。

・OSデータにおける作業領域枯渇による待ち。

例) 主記憶装置と2次記憶装置との間でのデータ入れ換え、各種バッファメモリの管理。

・プロセス（タスク）間のコミュニケーション（計算機内外）の事象待ち。

例) セマフォ、タスク間メッセージ通信、計算機間通信。

・CPUの利用制限による待ち。

例) プロセス（タスク）の優先度による実行待ち、或いは時間制限による実行停止。

・自発的な待ち。

例) プログラム自身によるタイマ機能を利用した一定時間の停止。

【0040】本発明におけるシステム管理モニタ3では、特定及びシステム全体で動作するタスク4において、停止原因別の停止時間の度数分布を一定のインターバル毎に提供する。図5は、本実施の形態2によって提供される停止原因別の停止時間の度数分布の概略図を示している。図5において、縦軸はインターバル内のタスク4の停止回数、横軸は1回の停止あたりの停止時間を表しており、この実施の形態2では停止時間を10m秒で丸めて報告している。図5に示す例では、注目した停止原因のほとんどが10～20m秒の間で終了するが、たまに停止時間が長い場合があることを示している。

【0041】図6は、本実施の形態2における、タスクの停止原因別の停止時間の度数分布表示を行うまでの処理の流れを示す流れ図で、以下図6を参照しながら本実施の形態2における動作について説明する。ステップS201（停止状況取得手段及び停止状況提供手段に相当）では、計算機システム上で実行中の全てのタスク4に関して、現在の停止状況を得る処理を行う。また、この時には、この停止状況を得た時刻も同時に記録する。なお、停止状況の取得は、OS2が提供するAPIの実行によるアトミックな処理であるが、OS2内部での処理については、後述の実施の形態4にて説明する。

【0042】次に、ステップS202において、ステッ

アS201で収集したスナップショットをシステム管理モニタ3が負荷状況表示を行うためのデータとして処理する。次に、ステップS203において、実施の形態1での処理において説明した図4の流れ図のステップS103, S107, S108, S113で書き出したログを処理するか否かを判断して、その結果により処理が異なる。もし、ログがない場合には、処理を行わない。しかしながら、ログを処理しない場合においても、計算機システム内のタスクの状況についてのスナップショット表示を得ることができる。

【0043】ログがあれば、ステップS204に移行して、未処理のログを1レコード読み出す処理を行い。次に、ステップS205に移行して、読み出したレコード(ログ)がインターバル内のデータとして処理するのに相応しいかどうかの判断を、ステップS201の処理で得た時刻情報を基にステップS205で行う。次に、ステップS206において、ステップS204で読み込んだレコード(ログ)を、負荷状況表示データとして処理する。次に、ステップS207では、未処理のログがあるかどうかを判断して、未処理のログが残っている場合には、ステップS204の処理に戻る。未処理のログが無ければステップS208に移行して、ステップS202及びステップS206において処理したデータを、停止原因毎の停止時間の度数分布図にして表示する。

【0044】以上のように本実施の形態2によれば、システム管理モニタが、一定のインターバル毎に、タスクの停止状況を基にタスクの停止原因毎の停止時間を度数分布にして表示するようにしたので、タスクの停止する原因を視覚的に捉えることができる。

【0045】実施の形態3。本発明におけるシステム管理モニタ3では、実施の形態例2において説明したタスク4の停止原因毎の停止時間に関する度数分布表示をインターバル内で得ると共に、累積期間でも得ることが可能である。本実施の形態3では、本発明におけるシステム管理モニタ3において、特定またはシステム全体で動作するタスク4について、累積期間で停止原因別の停止時間についての度数分布の提供について説明する。

【0046】図7は、本実施の形態3における、累積期間における停止原因毎の停止時間の度数分布表示を行う処理の流れを示す流れ図であり、以下、この図7に示す流れ図を参照しながら説明する。本実施の形態3での処理は、実施の形態2での説明に用いた図6の処理の流れとほとんど同じであり、処理開始時の処理方法が異なるだけである。即ち、実施の形態2における図6に示すステップS202で処理して度数分布表示に用いたデータは、測定インターバル(システム管理モニタ3の動作インターバル)の内だけで有効なので、次の測定インターバルでデータを積算する時の処理では、このデータを無効化する必要がある。これをステップS300として、この無効化の処理を処理開始時に実行する。また、ステ

ップS300の処理でデータの無効化が実現できるよう、度数分布表示を行うことと、新たな測定インターバルで得た停止情報を前回の測定インターバルで得ていた情報と重ね合わせ処理が必要となり、この処理をステップS308(実施の形態2におけるステップS208の処理に相当)で行う。

【0047】以上のように本実施の形態3によれば、システム管理モニタが、累積期間におけるタスクの停止原因毎の停止時間の度数分布表示を行うようにしたので、タスクの負荷状況をある程度長いレンジで確認することができる。

【0048】実施の形態4。本実施の形態4では、OS2がタスク4の停止原因とその停止時間を提供するAPI手段について説明する。本発明に係わるAPIでは、実行を再開したタスク4における停止原因とその停止時間を提供するのみではなく、現在も停止中のタスク4での停止開始からの停止時間をも提供する。本実施の形態4で説明する手段を用いることにより、長時間停止しているタスク4を検出することが可能になる。

【0049】図8は、本実施の形態4における、OS2がタスク4の停止状況を提供する手段の処理の流れを示す流れ図であり、以下に、この図を参照しながら説明する。先ず、ステップS401において、プロセッサに対しての割り込み処理を禁止するための処理を行う。これは、図8に示す一連の処理を短時間にアトミックに行うための処置である。

【0050】次に、ステップS402において、タスク管理テーブル20へのアクセスを禁止する処理を行う。この処理は、計算機システムが密結合方式のマルチプロセッサシステムの場合、ここで処理によって、タスク管理テーブル20が他のプロセッサによって書き換えられることを防ぐための処理である。

【0051】次に、ステップS403において、本処理を実行した時の時刻データを収集して、ステップS404において、ステップS403で収集した時刻データとタスク4のスナップショット情報を、本処理を要求したタスク(システム管理モニタ3)に渡す。タスク4のスナップショット状態は、タスク管理テーブル20上の、タスクの状態28と最後に停止した時刻21と最後に停止した原因22を基にした、停止中のタスク4における停止原因と停止開始からの時間情報である。

【0052】次に、ステップS405において、ステップS402でロック(禁止)していたタスク管理テーブル20へのアクセス禁止を解除する処理を行う。最後に、ステップS406において、ステップS401で禁止していたプロセッサへの割り込みを許可する処理を行う。

【0053】以上のように本実施の形態4によれば、システム管理モニタがタスクの停止状況を収集するときに停止中のタスクについての、その停止状況を取得するこ

とができるので、長期間停止中のタスクを検出することができる。

【0054】実施の形態5. 本実施の形態5では、本発明によるシステム管理モニタ3において、計算機システムの異常状態の検出、その検出結果によりボトルネックを検出する手段について説明する。本発明ではシステム管理モニタ3において、計測したタスクの停止時間に関して、下記に示す(1)～(3)の少なくとも一つによつて、異常事態を定義するフィルタを設定することができる。

- (1) 一度の停止時間が許容値を超えた。
- (2) 停止原因別の停止時間が許容値を超えた。
- (3) 停止時間の累積が許容値を超えた。また、フィルタの設定によって異常状態だと判断をした後には、システム管理モニタ3は、利用者が望むユーザ定義タスク4を投入することで、異常状態の記録、報告、回避、回復等の処理を自動化できる。

【0055】図9は、本実施の形態5における、システム管理モニタ3の有する異常状態検出手段についての処理の流れを示す流れ図であり、以下、この図を参照しながら説明する。なお、システム管理モニタ3は、図9に示す処理を、一定時間毎に実行する。先ず、ステップS501において、異常状態の定義が変更されているか否かを判断する。判断の結果、異常状態の定義が変更されている場合には、ステップS510に移行してシステム管理モニタ1/F手段により、ユーザが定義した異常状態を設定する。

【0056】異常状態の定義が変更されていない場合には、ステップS502に移行して、実施の形態2または実施の形態3で説明した停止状況収集手段3aで収集した計算機システムの停止状況(負荷状況)を表すデータを参照して、異常状態と定義された事態になっているかどうかの診断を行う処理を行う。次に、ステップS503において、ステップS502で計算機システムの診断を行った結果についての判断を行い、異常と判断する状態があった場合には、ステップS520でユーザが定義したタスク4に異常状態通知手段3dにより報告する。または、発生した異常状態に対処するユーザ定義タスク4をタスク投入手段3fにより起動して、問題解決にあたる。

【0057】以上のように本実施の形態5によれば、システム管理モニタが、計算機システムの異常状態を検知し、異常状態から対処することをタスク自身に通知するように、あるいは、適切なタスクを起動するようにしたので、異常状態に対して適切な対処をすることができる。また、例えば、異常状態の発生する原因を調査することにより、システムのボトルネックがどこにあるかを知ることができる。

【0058】実施の形態6. 本実施の形態6では、本発明によるシステム管理モニタ3において、タスク4の状

態遷移を把握することが可能なトレース情報を提供する手段について説明する。図10及び図11は、本実施の形態6を説明する図であり、図10は、本実施の形態6により提供されるタスクの遷移状態図(トレース図)、図11は本実施の形態6におけるシステム管理モニタ3の処理の流れを示す流れ図である。

【0059】本発明によるシステム管理モニタ3では、停止状況収集手段3aで収集したデータを基にタスク4の状態遷移をタスク状態遷移表示手段により、図10のようなトレース図(遷移状態図)を提供する。図10の縦軸はタスクの状態を表しており、本発明においては、実行状態、CPU待ち状態、停止状態の3状態がある。横軸は時間軸を表している。システム管理モニタ3は、実施の形態2または3においてステップS206で説明したログ処理でのデータを元に、図10に示すグラフを作成する。このグラフの元データは、状態遷移した時間、遷移前後の状態、タスクの識別子である。遷移した時間の場所で、遷移前から遷移後の状態に向けて矢印を引く。矢印の下には、遷移前後の停止原因を記述する。

【0060】図10について補足説明をすると、図における[-]は、"実行状態"を、[CPU]は、"CPU待ち状態"を、[ipc:internal process communication]は、停止の一原因でタスク間通信による"停止状態"を示している。例えば、図10に示されている時系列の最初の状態は、"タスク1"が"実行状態"から"CPU待ち状態"に遷移したことを、次の状態は、"タスク2"が"停止状態"から"CPU待ち状態"に遷移したことを示している。

【0061】次に、本実施の形態6におけるシステム管理モニタ3における処理の流れを示す図11の流れ図を用いて説明する。ステップS204からステップS207までの処理は、実施の形態2または実施の形態3での同ステップ番号の処理と同じである。本実施の形態6に特有の処理は、ステップS601での処理(状態遷移表示手段)を新たに追加し、図10に示したようなトレース図を表示することである。

【0062】以上のように本実施の形態6によれば、システム管理モニタが、タスクの状態遷移を把握することが可能なトレース情報を提供するようにしたので、マルチタスク実行方式の計算機において、タスク間の依存関係を把握することが容易になる。

【0063】実施の形態7. 本実施の形態7では、本発明によるシステム管理モニタ3において、計算機システムに投入するタスク4を制御して、タスク4のループットを向上させる手段について説明する。本発明によるシステム管理モニタ3は、タスク投入手段3fを備えている。ユーザ定義タスク4が本発明によるシステム管理モニタ1/F手段5を用いて、投入するタスク情報を取得して、タスク投入手段3fにより計算機システムにタス

ク4を投入するようにしている。このようにすることにより、システム管理モニタ3がタスクの停止状況収集手段3aで収集したタスクの停止状況、即ち計算機の負荷状況に基づいて、計算機システムにタスク4を投入することを可能としている。従って、一度に複数のタスク4を計算機システムに投入しないようにして、OS2のオーバーヘッドを減らす、即ち、タスク4のスループットを向上させる効果がある。

【0064】図12(a)及び(b)は、このタスク投入手段の処理の流れを示す流れ図であり、図12(a)は、ユーザタスク4での処理を、図12(b)は、システム管理モニタ3での処理を示している。本実施の形態7においては、タスク4の停止時間を基にした負荷状況を判断して、その結果によりタスクを投入するべきかどうかを決定する。以下、図13に示した流れ図を参考しながら説明する。

【0065】ユーザ定義タスク4においては、ステップS701で、システム管理モニタ3に投入するタスクの情報を渡すために、システム管理モニタ1/F手段5に、その情報を設定しておく。実際にタスクを投入する処理は、システム管理モニタ3に任せること。このような場合に必要になる、異なるタスク間での情報伝達手段は、当業者には公知の方法が種々あるが、その方法についての、ここでは触れない。

【0066】図12(b)に示すステップS710からステップS713までは、システム管理モニタ3側での、タスク投入手続きを示している。システム管理モニタ3がこの一連の処理を行うのは、タスク投入側によってステップS701が実行された時や前回の処理から一定時間経た後等、計算機システムに対して新たなタスク4の投入が必要になった場合である。

【0067】ステップS710(タスク投入可否判断手段)において、計算機システムにタスク4の投入を行うべきかどうかを判断する。例えば、CPU待ちの時間が多いため等により、OS2のオーバーヘッドが増大している状態では、新たなタスク4を投入するのに相応しくないと判断し、処理を抜ける。なお、タスク投入時期の善し悪しの判断は、実施の形態5に示したあげた診断ルーチンと同様な方法による。タスクの投入が可能と判断すると、ステップS711において、ステップS701の実行で渡された、投入するタスクに関する情報をシステム管理モニタ1/F手段5から得る処理を行う。次いで、ステップS712において、計算機システムにタスクを投入する処理を行う。次に、ステップS713で、他にペンドィング中のタスク投入処理があるかどうかを判断して、ペンドィング中のものがある場合に、ペンドィング中のタスク投入要求に対してステップS710からの処理を繰り返す。

【0068】以上の処理のように、本実施の形態7によれば、システム管理モニタを通してタスク投入を行うよ

うにしたので、計算機システムの負荷状況に応じて投入するタスクを制限することが可能となり、タスクのスループットを向上させることができることになる。

#### 【0069】

【発明の効果】以上のように、この発明によれば、タスクの停止時間の許容値を設定する手段を設けると共に、OSがタスク毎にタスクの停止原因と停止時間を管理するようにしたので、タスクの停止時間が許容値を超えた場合には、OSがタスクに対して、事象通知を行うことができる。従って、タスク自身が、タスクの実行に遅れが生じたことを認識することができるので、タスク自身で処理の遅れに対処する処理等の回復処理を自動化させることができる。

【0070】また、所定の周期で動作するシステム管理モニタを設けて、このシステム管理モニタがタスクの停止状況、即ち負荷状況を度数分布表示するようにしたので、タスクの負荷状況を視覚的に捉えることができる。

【0071】また、タスクの停止状況の表示をその停止原因別に所定のインターバル毎に表示するようにしたので、突発的に発生する停止原因の把握をすることができる。

【0072】また、タスクの停止状況の表示をその停止原因別に累積値で表示するようにしたので、タスクの全体的な停止状況を把握することができる。

【0073】また、停止状態中のタスクの情報も取得できるようにしたので、長期間停止しているタスクの検出が可能となり、その停止原因に対する処理を行うことが可能となる。

【0074】また、システム管理モニタにタスクの異常状態を定義する手段を設け、定義した異常状態が発生すると、タスクに対してその報告をするようにしたので、タスク自身で、以後の対処が可能となる。

【0075】また、システム管理モニタは、異常状態が発生すると、その異常状態に対処するタスクを自動的に起動するようにしたので、異常処理を自動的に行うことができる。

【0076】また、異常状態の定義を更新する手段を設けるようにしたので、システムの運営を柔軟に行うことができる。

【0077】また、システム管理モニタに、タスクの状態遷移を表示する手段を設けるようにしたので、タスク間の依存関係を把握することができ、この依存関係を解析することによりシステムとして最適なスループットが得られるようにタスクの運用を行うことが可能となる。

【0078】また、タスクの負荷状況を把握しながら、タスクを投入するようにしたので、システムのスループットの向上を図ることができる。

【0079】また、予めタスクの停止時間の許容値を設定しておいて、タスクの停止時間を計測して、停止時間が許容値を超えると、タスク自身に報告するようにした

ので、タスク自身で対策を行うことが可能となる計算機システムを提供することができる。

【0080】また、タスクの停止状況を度数分布で表示するようにしたので、タスクの負荷状況を直感的に把握することのできる計算機システムを構築することができる。

【0081】また、タスクの異常状態を定義して、定義した異常状態が発生するとこの異常状態に対処するようにしたので、自動的に異常処理を行う計算機システムを提供することができる。

【0082】また、異常状態に対する処理をタスク自身で行うようにしたので、最適な対処をすることができる。

【0083】また、異常状態に対する処理を、予め対応付けておいてタスクにより行うようにしたので、自動的な対処をすることができる。

【図面の簡単な説明】

【図1】 本発明によるタスク管理装置のソフトウェアの全体構成を示す図である。

【図2】 実施の形態1におけるタスクの状態遷移を時系列に示した図である。

【図3】 実施の形態1におけるタスク管理テーブルとトレースバッファの詳細を示す図である。

【図4】 実施の形態1におけるタスクの停止時間管理の処理の流れを示す流れ図である。

【図5】 実施の形態2におけるシステム管理モニタが表示するインターバル毎の度数分布の例を示す図である。

【図6】 実施の形態2におけるシステム管理モニタの

行う処理の流れを示す流れ図である。

【図7】 実施の形態3におけるシステム管理モニタの行う処理の流れを示す流れ図である。

【図8】 実施の形態4におけるAPI手段の行う処理の流れを示す流れ図である。

【図9】 実施の形態5におけるシステム管理モニタの有するボトルネック検出手段の行う処理の流れを示す流れ図である。

【図10】 実施の形態6におけるシステム管理モニタの表示する状態遷移図の一例を示す図である。

【図11】 実施の形態6におけるシステム管理モニタの行う処理の流れを示す流れ図である。

【図12】 実施の形態7におけるシステム管理モニタの有するタスク投入手段の行う処理の流れを示す流れ図である。

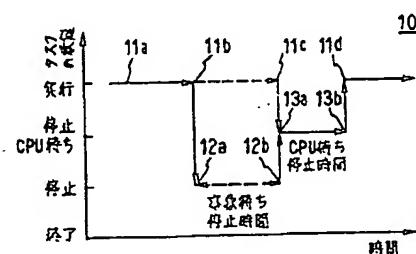
【図13】 UNIXにおけるpsコマンドの出力形式を示す図である。

【図14】 UNIXにおけるvmstatコマンドの出力形式を示す図である。

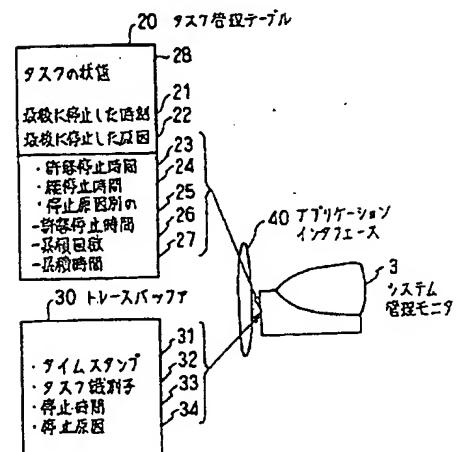
【符号の説明】

1 ソフトウェア構成図、2 オペレーティングシステム、2a 停止許容時間設定手段、2b 停止時間累積手段、2c 停止状況提供手段、2d 停止時間超過報告手段、3 システム管理モニタ、3a 停止状況収集手段、3b 停止状況表示手段、3c 異常状態検出手段、3d 異常状態通知手段、3e タスク状態遷移表示手段、3f タスク投入手段、4 ユーザ定義タスク、5 システム管理モニタI/F手段、20 タスク管理テーブル、30 トレースバッファ。

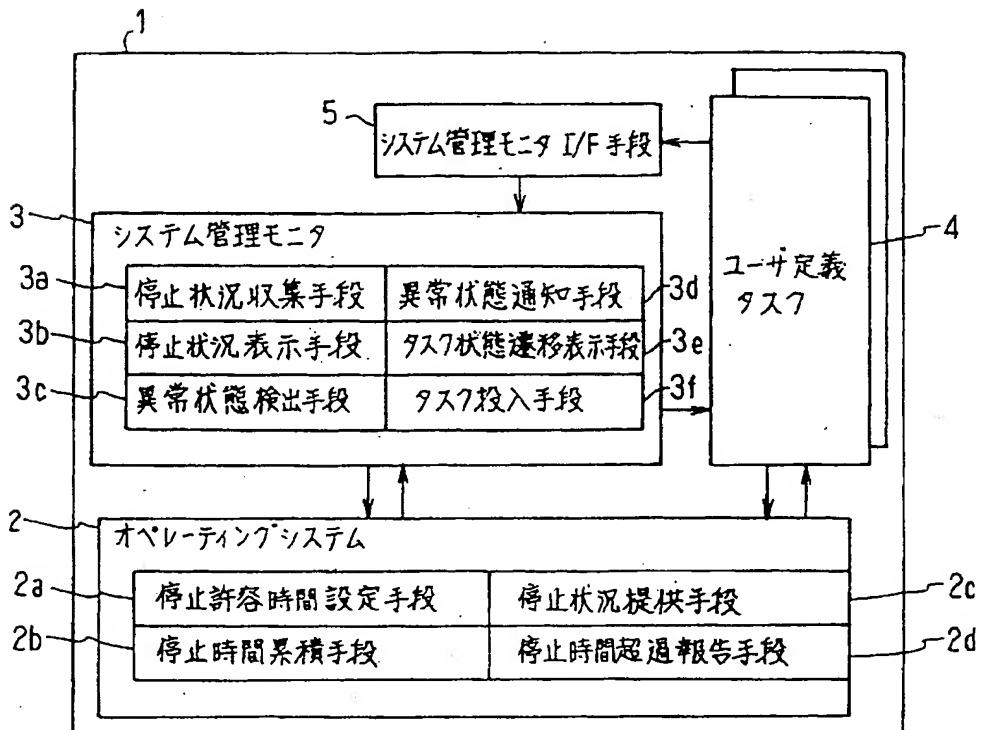
【図2】



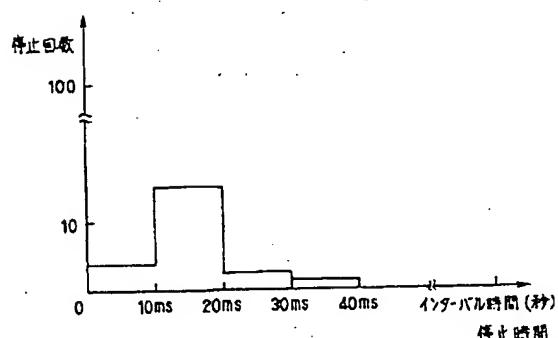
【図3】



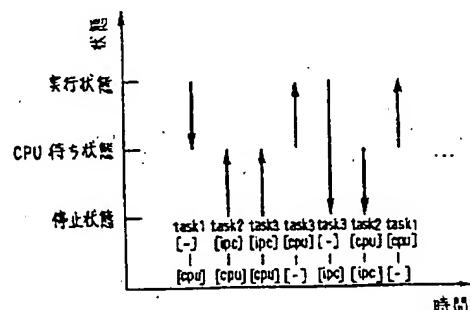
〔 1〕



〔図5〕



【图10】

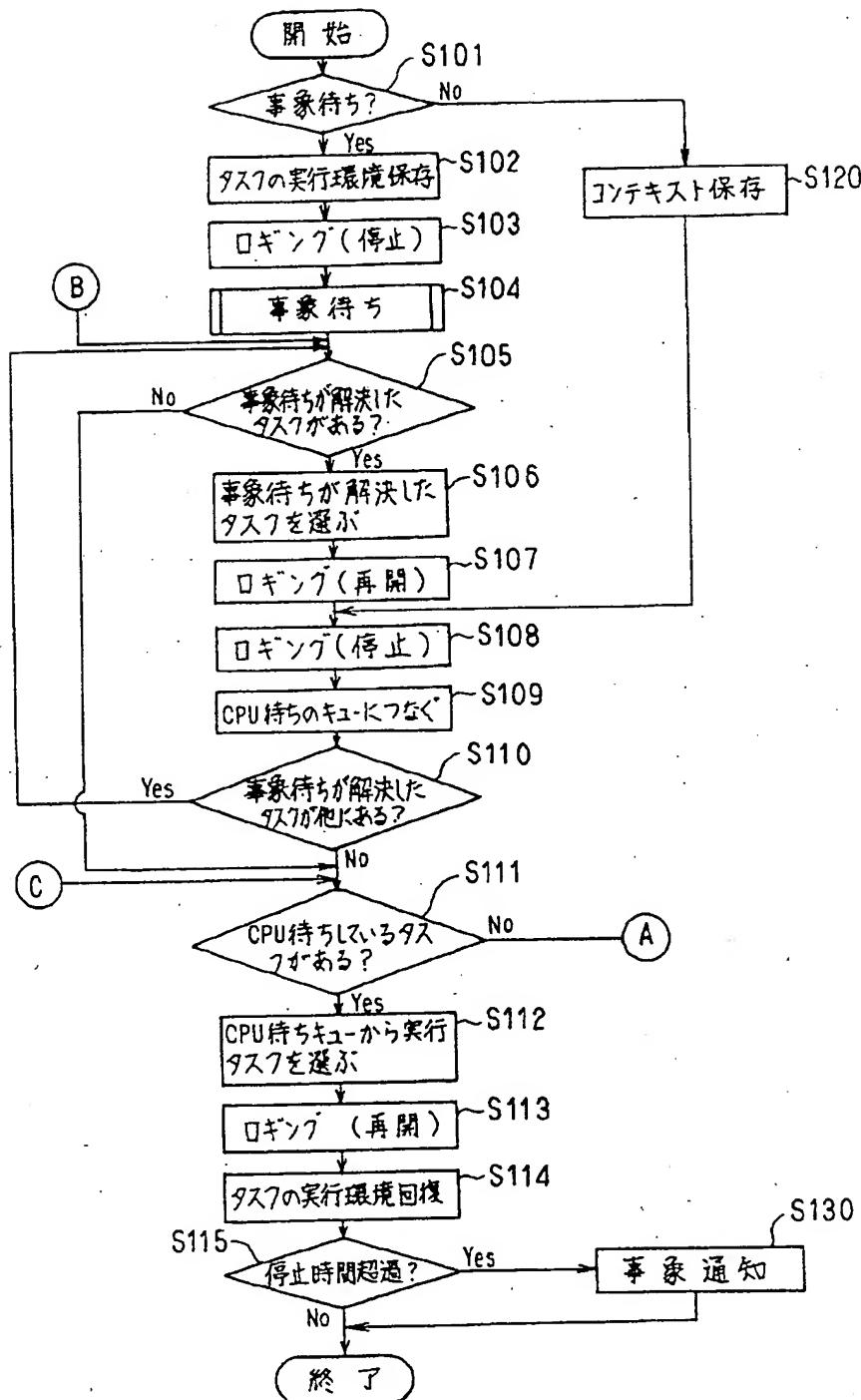


【图13】

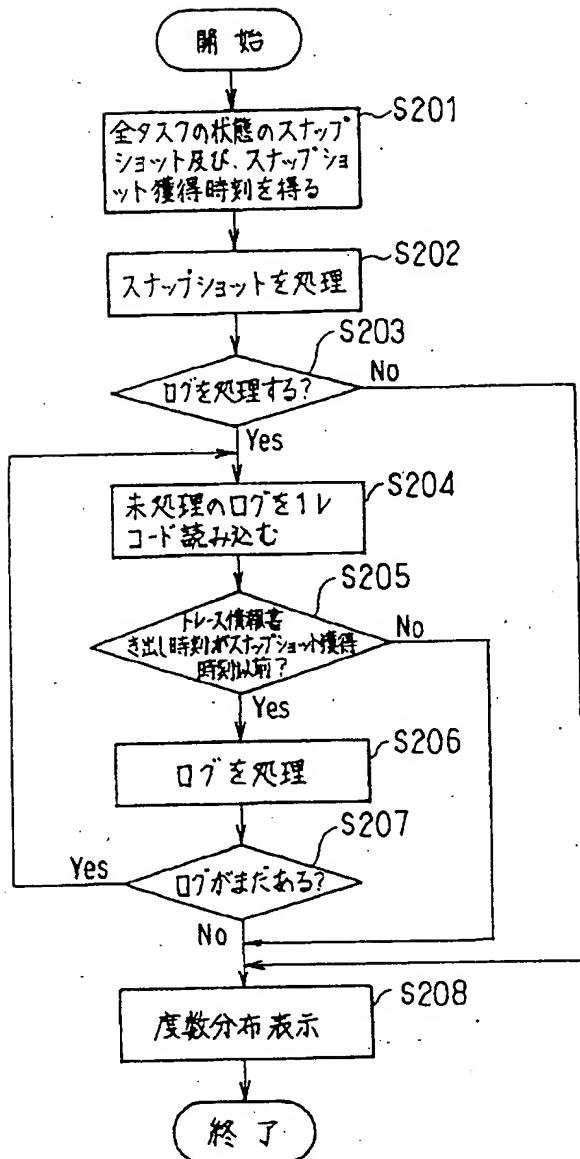
F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	COMMAND
3	S	root	0	0	0	128	20	106758	0		Jan 1	?	0:14	swapper
0	S	root	1	0	0	168	20	305f540	54	7ffe6000	Aug 17	?	0:00	init
3	S	root	2	0	0	128	20	3051480	0	1e4580	Aug 17	?	0:03	vhand

```
1 S  root 125 100 6 154 20 31ee600 17 323de8 14:12:09 ttys1 0:02 xterm
1 R  root 123 100 3 178 20 31eeb00 19 14:52:25 ttys1 0:00 ps -lf
```

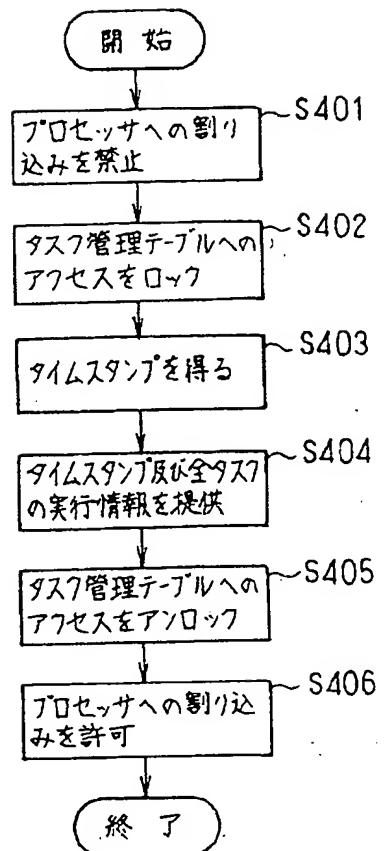
【図4】



【図6】



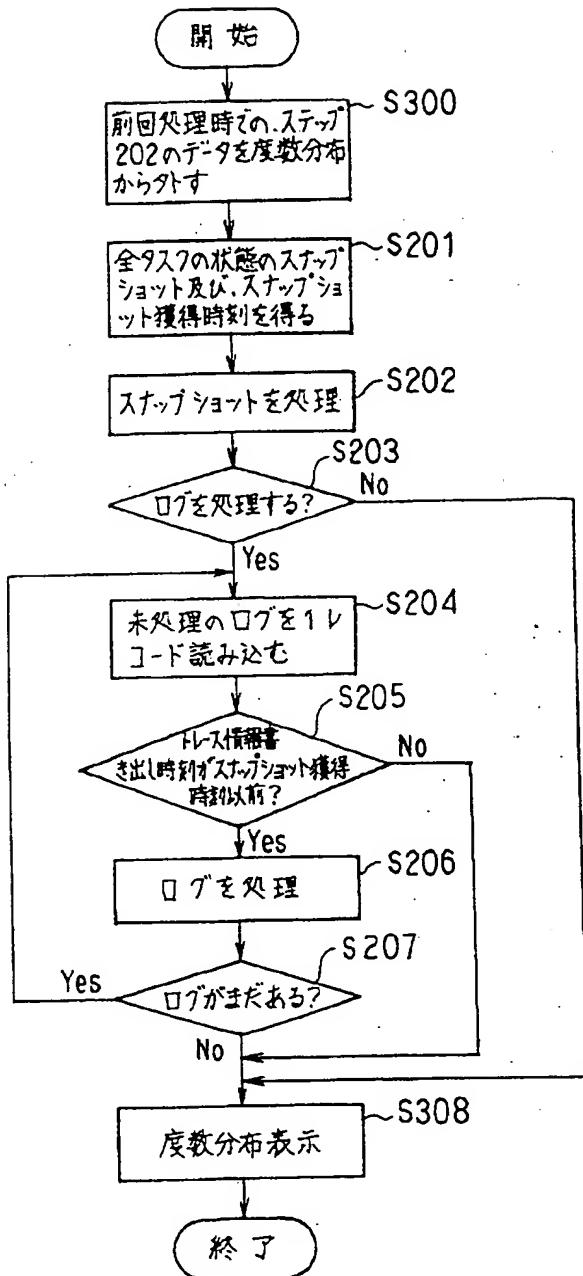
【図8】



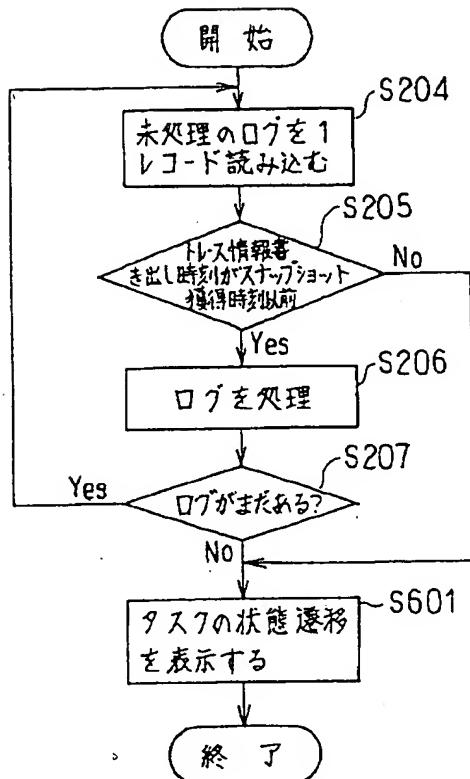
【図14】

procs	b	w	avm	free	re	st	pi	po	fr	de	sr	in	sy	cs	us	sy	id	
0	0	12	3243	244	0	0	0	0	0	0	0	0	40	61	15	0	0	100
0	0	15	2536	223	0	0	0	0	0	0	0	0	78	96	31	0	0	100
0	0	15	2536	223	0	0	0	0	0	0	0	0	54	46	18	0	0	100
0	0	14	2477	226	0	0	0	0	0	0	0	0	51	36	14	0	0	100
0	0	14	2022	226	0	0	0	0	0	3	0	37	26	12	0	0	100	

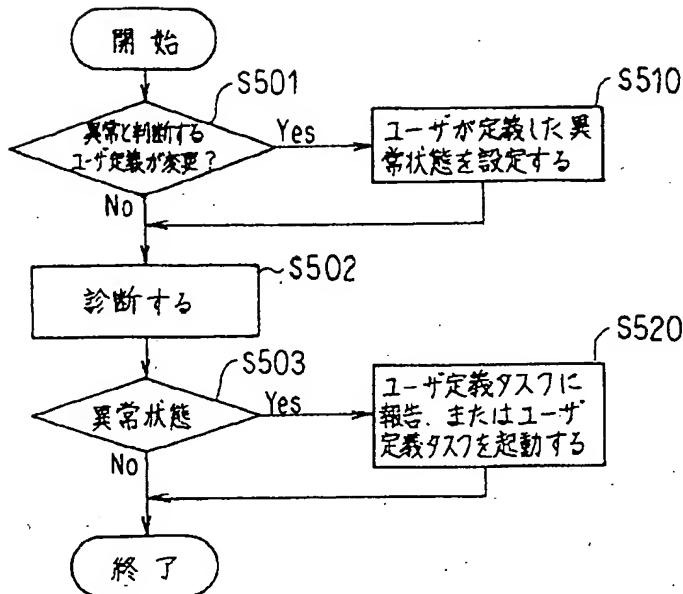
【図7】



【図11】

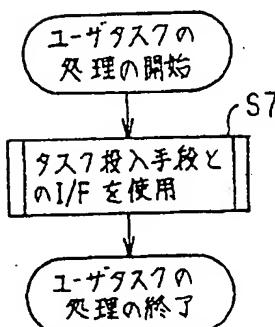


【図9】



【図12】

(a)



(b)

